

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application. There are currently no amendments to the claims.

**Listing of Claims**

1. (Previously Presented) A method of modifying properties of a lock associated with a resource in a distributed environment, wherein the lock has a lock owner, the method comprising:
  - receiving a request to modify at least one property associated with the lock, wherein the request originates from a requesting client computer system;
  - analyzing the request to determine whether the request is made by the lock owner; and
  - if the request is made by the lock owner, modifying the at least one property.
2. (Original) The method as defined in claim 1 wherein the method further comprises:
  - following the determination of whether the request is made by the lock owner, determining whether the resource is locked by another client computer system that may conflict with the requested modification; and
  - if the resource is locked by a conflicting lock, denying the received request.
3. (Previously Presented) A method as defined in claim 1 wherein the request relates to modifying a lock type property.
4. (Previously Presented) A method as defined in claim 1 wherein the request relates to the modification of the lock scope property.
5. (Previously Presented) A method as defined in claim 1 wherein the request relates to the modification of a lock ownership.
6. (Original) A computer program product readable by a computer and encoding instructions for executing the method recited in claim 1.

7. (Original) A computer program product readable by a computer and encoding instructions for executing the method recited in claim 5.

8. (Original) A computer-readable medium having stored thereon a locked resource, wherein the locked resource comprises:

a resource object data section for storing actual object data;

a lock object, wherein the lock object comprises a plurality of properties, wherein a first property identifies a lock owner, and wherein the properties may be modified by the lock owner.

9. (Original) A computer-readable medium as defined in claim 8 wherein a second property relates the resource object and wherein the second property may be modified by the lock owner to associate the lock object with a second resource object.

10. (Original) A computer-readable medium as defined in claim 8 wherein the lock owner may modify the first property relating to lock ownership to transfer the lock object to a second owner.

11. (Previously Presented) A system for managing access of one or more resources by a plurality of processes in a distributed environment, the system comprising:

a receive module for receiving resource requests from the plurality of processes, wherein the receive module receives a request from a requesting process that includes modification information concerning at least one property of a lock object associated with a requested resource;

a determination module operable to determine whether the requesting process owns the lock object; and

an update module operable to modify the at least one property of the lock object as set forth in the modification information upon a determination that the requesting process owns the lock object.

12. (Previously Presented) A system as defined in claim 11 wherein the determination module also determines whether there is a conflicting lock associated with the

requested resource and wherein the update module does not modify the at least one property of the lock object upon a determination that a conflicting lock exists.

13. (Previously Presented) A system as defined in claim 11, wherein the lock object has a lock type property, and wherein the update module modifies the lock type property as set forth in the modification information.

14. (Previously Presented) A system as defined in claim 11, wherein the lock object has a lock scope property, and wherein the update module modifies the lock scope property as set forth in the modification information.

15. (Previously Presented) A system as defined in claim 11, wherein the lock object has a lock ownership property, and wherein the update module modifies the lock ownership property as set forth in the modification informationu to thereby transfer the lock object from one process to another.

16. (Original) A system as defined in claim 11 further comprising a transfer module for transferring ownership of the lock object from the requesting process to another process.

17. (Original) A system as defined in claim 11 wherein the requesting process communicates with receive module using Web Distributed Authoring and Versioning protocol.

### **REMARKS/ARGUMENTS**

This Amendment and Response and the following remarks are intended to fully respond to the Office Action mailed February 16, 2006. In that Office Action, claims 1-17 were examined, and all claims were rejected. More specifically, claims 1-16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,510,478 to Jeffords et al. (hereinafter, "Jeffords"), in view of U.S. Patent No. 6,026,401 to Brealey et al. (hereinafter, "Brealey"); and claim 17 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Jeffords in view of Brealey, in further view of Applicant's admitted prior art. Reconsideration of these rejections is respectfully requested.

In this Response, no claims have been amended, canceled or added. Therefore, claims 1-17 remain present for examination.

#### **Claim Rejections – 35 U.S.C. § 103**

Claims 1-16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Jeffords in view of Brealey. Applicant respectfully traverses the § 103(a) rejections because the Examiner has failed to state a prima facie case of obviousness. A prima facie case of obviousness can be established only when all of the following requirements are satisfied: (1) the reference or combination of references must teach or suggest all of the claim limitations; (2) there must be some suggestion or motivation in the references themselves to combine the references; and (3) there must be a reasonable expectation of success. See MPEP §§ 706.02(j) & 2143. Thus, the combination of references cited by the Examiner must teach or suggest every limitation of the claimed invention. CFMT, Inc. v. YieldUp Int'l Corp., 349 F.3d 1333, 1342 (Fed. Cir. 2003); see also MPEP § 2143.03.

The present invention relates generally to managing the use of resources by a plurality of processes in a distributed environment. In particular, the invention relates to a system and methods of locking distributed environment resources through the use of "locks," or "lock objects," to prevent inappropriate access to such resources. A lock has certain properties associated with it. Such properties include, for example, the ownership, scope, type, and/or resource association, of the lock. These properties of the lock may be modified, or changed, without requiring release of the lock to accomplish such modifications to it. For example, where

the type or scope properties of the lock are, for example, “exclusive” or “shared,” “read lock” or “write lock,” “mandatory” or “advisory,” etc., modifying lock properties relates to “modify[ing] the scope or type properties of the lock object 306 from an exclusive lock to a shared lock, or from a read lock to a write lock, or from a mandatory lock to an advisory lock” (emphasis added). The ownership property of a lock may also be modified in an embodiment. In these embodiments, the present invention relates to the modification of the properties of the lock itself, and thus, not to modifications to the resource which is locked by the lock. Each of the claims is directed to providing a lock owner with the ability to modify the properties of a lock without being required to release the lock.

Jeffords relates generally to the granting and releasing of locks on resources for requesting processes in a distributed environment but does not refer to properties of any kind in the context of such locks. In other words, Jeffords refers only to “locks” in general and does not disclose any properties of such locks, e.g., shared/exclusive, advisory/mandatory, read/write, much less does not disclose any modifying of such properties as claimed and disclosed by the present invention.

Similarly, Brealey relates to the use of locks for managing data which is commonly accessible, but it does not teach the modifying of those locks. Brealey teaches the lock properties of shareable locks with read-only access and exclusive locks which allow the user to modify the locked data. Further, Brealey teaches that such locks may be destructed and constructed. However, Brealey fails to teach that such locks may be modified. Constructing and destructing locks is not the same thing as modifying a lock. Construction and destruction deals with creation and deletion and, thus, not with changes to the lock properties themselves. If a certain type of lock property is desired, Brealey teaches that the current lock with the undesired property, if any, is destroyed, and a new lock with the desired property is constructed. Brealey’s discussion of modifications occurs only in the context of changes made to the underlying model objects and, thus, does not teach modifications of the locks controlling access to those model objects.

Jeffords and Brealey fail to satisfy the first prong of establishing a prima facie case of obviousness because they fail to teach or suggest all of the claim limitations. Claim 1 of the

present invention requires, among other elements, a “request to modify at least one property associated with the lock . . .” (emphasis added). Similarly, claim 8 of the present invention requires, among other elements, “a lock object, wherein the lock object comprises a plurality of properties, wherein a first property identifies a lock owner, and wherein the properties may be modified by the lock owner,” (emphasis added), and claim 11 requires, among other elements, “a receive module for receiving resource request from the plurality of processes, wherein the receive module receives a request from a requesting process that includes modification information concerning at least one property of a lock object associated with a requested resource” (emphasis added). As can be seen, each of these elements of independent claims 1, 8, and 11 requires that the properties of a lock, or lock object, be capable of modification. However, Jeffords and Brealey fail to teach or suggest the modification of a lock. Indeed, the Examiner has expressly stated that “Jeffords does not explicitly teach modifying at least one property associated with the lock.” Examiner’s Response to Amendment, Office Action, 2/16/2006, p. 3.

Further, contrary to the Examiner’s arguments, Brealey, like Jeffords, fails to teach or suggest the modification of a lock. First, it is important to clarify the meaning of the terms “lock” and an “underlying model object.” Brealey teaches that a “lock object locks access to the underlying lockable model object.” Brealey, col. 8, lines 47-48 (emphasis added). Thus, there is both a “lock object” and an “underlying model object.” These are two discrete entities. The model object is the entity that is locked by the lock object. The term “model object” is thus analogous to the present invention’s use of the term “resource.” Thus, in Brealey, a “lock” acts upon a “model object,” or resource, in that access to a model object is locked, or controlled, by a lock. The underlying model object and the lock controlling access to such model object thus cannot be the same entity. By analogy, access to a model object is locked by a lock, just as access to a house is locked by a padlock. Thus, a lock and its underlying model object, or resource, are not the same thing.

Brealey does refer to lock properties, e.g., “shared lock: shared read only access” and “exclusive lock: restricts read/write access to one process.” Col. 8, lines 44-45. However, Brealey in no way teaches the modifying of these properties of the lock as disclosed and claimed by the present invention. Taking an example with regard to the present invention’s teaching of

modifying a lock property, the present invention discloses, by way of example, that “an upgrade/downgrade module 324 [] may be used to change various properties within the lock object 306. For example, a process such as process A 312 may request to modify the scope or type properties of the lock object 306 from an exclusive lock to a shared lock . . .” Application, p. 15, lines 13-16 (emphasis added). Thus, the properties of the lock itself are changed or modified from an exclusive nature to a shared nature in an embodiment of the present invention. On the other hand, Brealey does not teach changing the properties of the lock itself. Instead, to change from an exclusive lock to a shared lock, Brealey teaches that the exclusive lock is destroyed and a shared lock is created:

[C]onstructing a lock object locks access to the underlying lockable model object. Destruction of the lock object is required to unlock access, in most cases. Thus, by instantiating a delete lock type to remove the instantiation of a shared lock type, shared restricted read/write access to the underlying model object is restored. This guarantees that the [underlying model] object will not be deleted, and allows others to read or modify it [i.e., the underlying model object] (that is to obtain a shared or exclusive modifying lock).

Brealey, col. 8, lines 48-54 (emphasis added).

Thus, to the extent Brealey suggests any “modifying,” it does so only with respect to modifying the underlying model object, or resource, and not with respect to modifying the properties of the lock itself. Indeed, Brealey teaches away from modifying a lock property and, instead, teaches the necessity of destructing and constructing a lock to change the type or scope of access. FIG. 9 of Brealey shows a lockable object, i.e., an underlying model object, which is first unlocked and then is locked upon the construction of a lock object. If modifications are made to the underlying model object, the underlying model object is now modified. Before the lock for that model object can be destructed, the modifications to the model object must be saved. While FIG. 9 and the discussion related thereto relate to modifying the underlying model object, at no point is the lock itself modified. Instead, the lock itself is only created and destroyed and is in no way modified. The discussion related to FIG. 9 states: “The exception is when the underlying locked object is modified. The lock can be destructed when the modification is saved.” Col. 8, lines 54-56 (emphasis added). Thus, the reference to modifications in

Brealey relates only to modifying the underlying model object, and does not relate to the lock controlling access thereto.

Further, the Examiner's reference to the Abstract to support the reading that "Brealey teaches modifying a lock property," Office Action, 2/16/06, at 3, is also misplaced. The Abstract provides, "Two types of lock objects are provided; shareable locks that may be shared by many user processes and permit only read access to the locked data, and exclusive locks, useable by only one user process at a time, that permit the owning process to modify the locked data." The Abstract makes no mention of modifying the properties of the lock, but instead treats the locks as two discrete, unchangeable types: shareable locks v. exclusive locks. FIG. 10 reinforces the separate nature of the lock versus the underlying model object, or lockable object. FIG. 10 shows the SharedLock and ExclusiveLock acting on a LockableObject, but the LockableObject, not the SharedLock or ExclusiveLock, is the entity which is modified by "int modify LockID." In contrast, the locks of the present invention are themselves changed, or modified, e.g., "to modify the scope or type properties of the lock object 306 from an exclusive lock to a shared lock." Application, p. 15 (emphasis added). The lock object, or lock, of the present invention is itself being modified, unlike the locks of Brealey which are first destructed and then constructed to change access types: "Destruction of the lock object is required to unlock access, in most cases." Col. 8, lines 48-50.

Jeffords does not mention properties in the context of locks, and while Brealey discusses lock properties, Brealey is completely silent and altogether fails to suggest, much less teach, any process related to the modifying of such properties of the locks. For at least these reasons, the Applicant respectfully requests reconsideration of the Examiner's rejection of claims 1, 8, and 11 in view of Jeffords and Brealey as these claims are believed to recite the present invention in a manner distinguishable over any combination of the above references. In addition, claims 2-7, 9-10, and 12-16 are also believed to be patentable over Jeffords in view of Brealey as these claims depend directly or indirectly from the allowable base independent claims 1, 8, and 11.



Further, Applicant notes that with respect to the Examiner's rejection of dependent claim 17 on the grounds that it is unpatentable over Jeffords in view of Brealey, in further view of Applicant's admitted prior art, claim 17 is believed to be patentable over the Examiner's cited art because claim 17 depends from what the Applicant believes is an allowable base claim 11. See discussion supra. The Applicant thus respectfully requests reconsideration of the rejection to claim 17 in light of the arguments and amendments presented above. Further, the Applicant notes that since the remarks above are believed to distinguish over the applied reference, any remaining arguments supporting the claim rejections are not acquiesced to even though they are not addressed herein.

### **Conclusion**

This Response fully responds to the Office Action mailed on February 16, 2006. It is recognized that the Office Action may contain arguments and rejections that are not directly addressed by this Amendment due to the fact that they are rendered moot in light of the preceding arguments in favor of patentability. Hence, the failure, if any, of this Amendment to directly address an argument raised by the Examiner should not be interpreted as reflecting the Applicant's belief that such argument has merit. Furthermore, the claims of the present application may include other elements, not discussed in this Response, which are not shown, taught, or otherwise suggested by the art of record. Accordingly, the preceding arguments in favor of patentability are advanced without prejudice to other bases of patentability.

It is believed that no further fees are due with this Response. However, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

In light of the above remarks and amendments, it is believed that the application is now in condition for allowance and such action is respectfully requested. Should any additional issues need to be resolved, the Examiner is requested to telephone the undersigned to attempt to resolve those issues.

Application No. 09/992,525

Respectfully submitted,

Dated: May 16, 2006



A handwritten signature in cursive script that reads "Elizabeth J. Reagan". The signature is written in black ink and is positioned above a horizontal line.

Elizabeth J. Reagan, Reg. No. 57,528  
MERCHANT & GOULD P.C.  
P. O. Box 2903  
Minneapolis, MN 55402-0903  
(303) 357-1644

K:\Clients\40\40062\0123-US-01\Resp & Amend to 2.16.06 OA.doc